

大促活动前团购系统 流量预算和容量评估实践

丁媛 2016-09-06

目录

大促活动前团购系统

流量预算和容量评估实践

01

背景介绍

02

流量预算

03

容量评估——压力测试策略

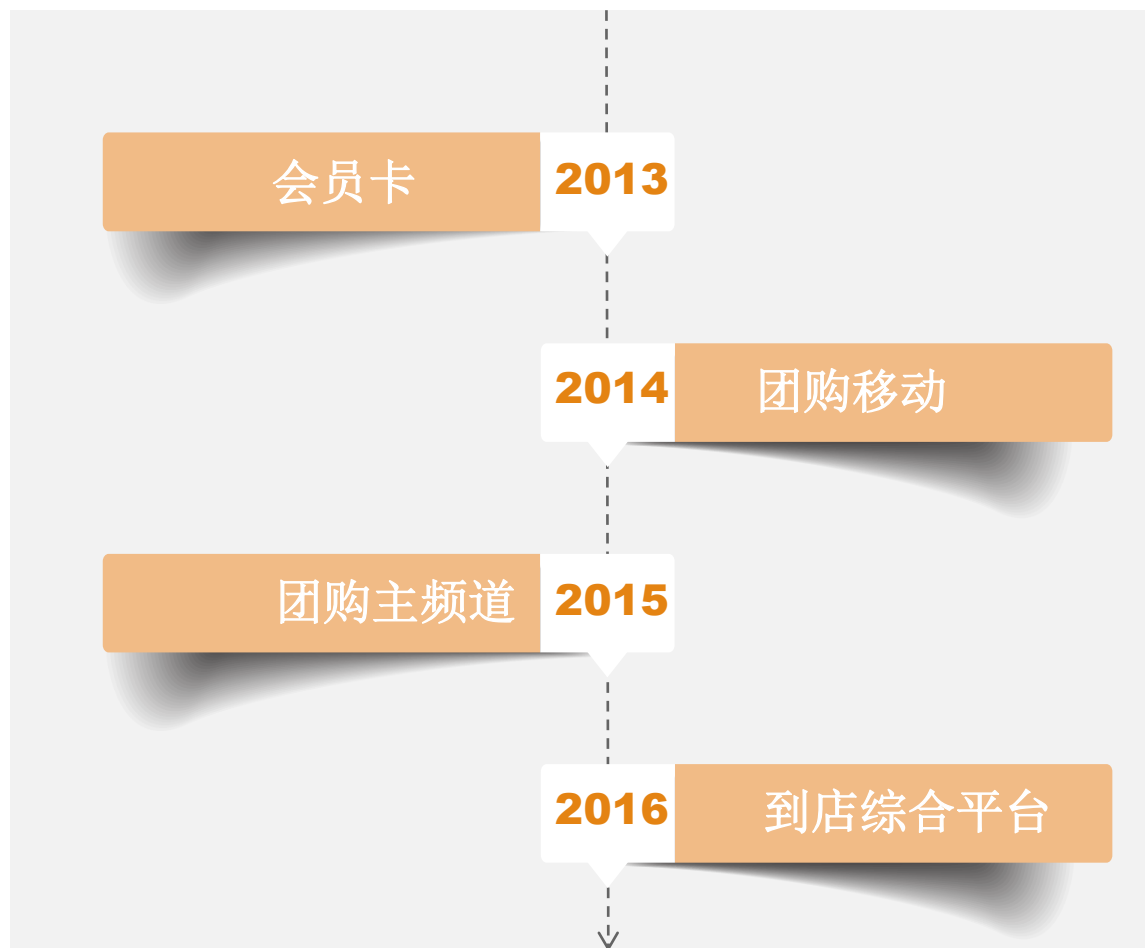
04

容量评估——压力测试方法

05

总结展望

个人介绍



丁媛

到店综合用户端测试组

综合平台

引言





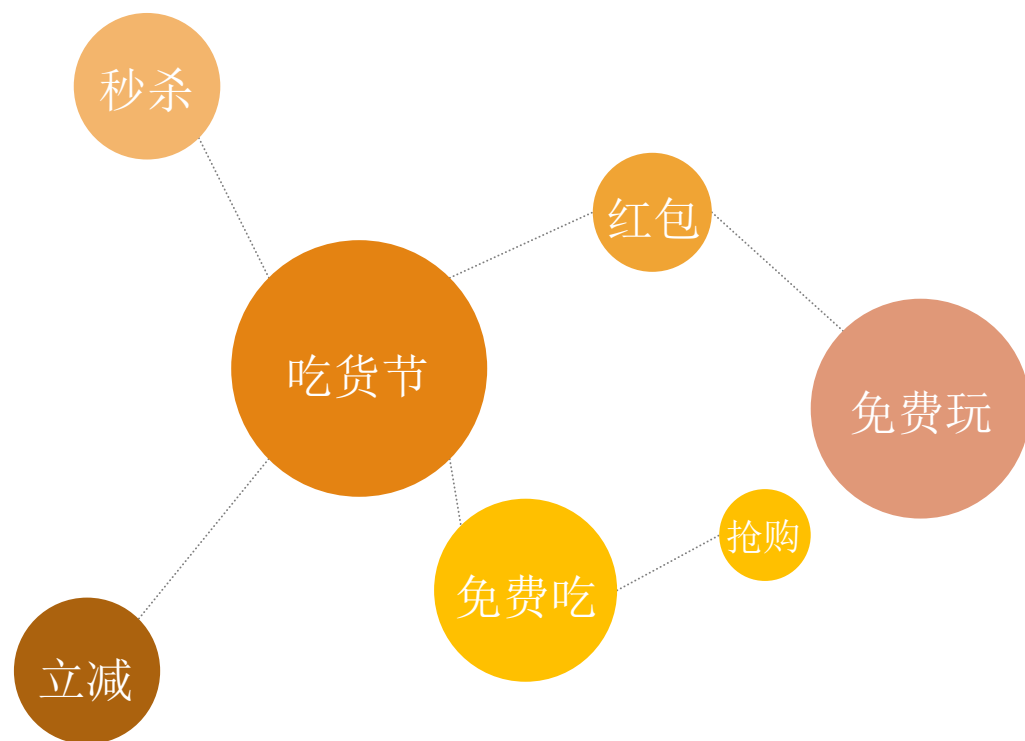
业务初期

重点是如何快速地进行功能开发和上线

2015年之后

产品形态比较稳定，产品运营会策划各种**大促活动**，为业务带来更多的流量和用户，从而提升交易额。

大促活动的特点

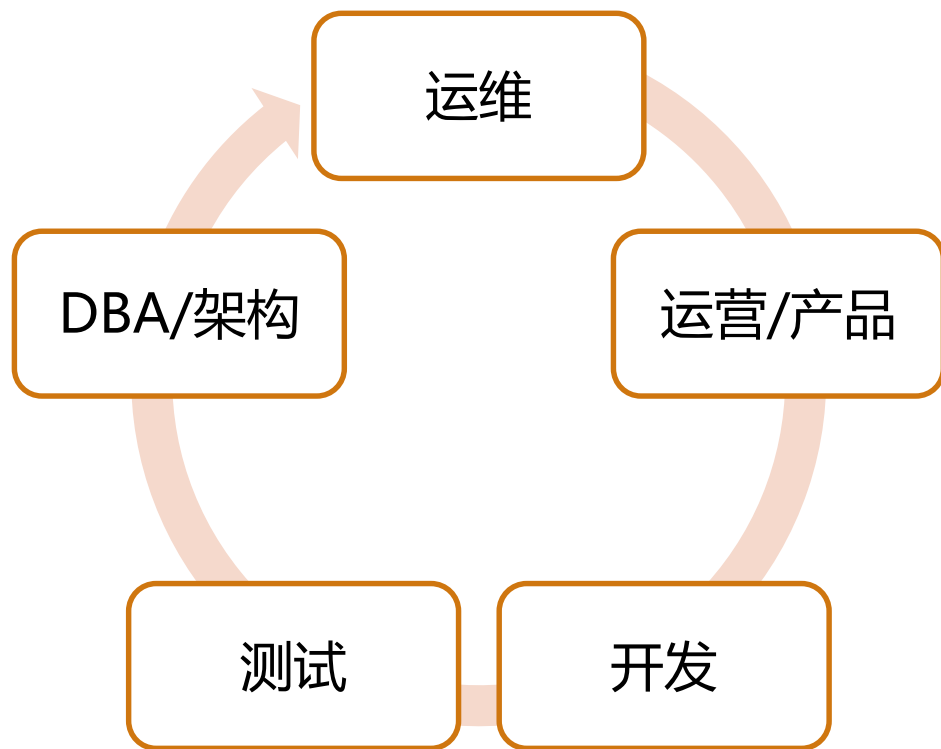


瞬时流量

核心路径

热点团单

大促活动前的准备



- 活动经费预算
- 活动push量
- 经验转化率



- 活动页的PV
- 活动页UV
- 抢购用户数



- 活动流量预算
- 系统容量评估

大促活动前的准备——扩容

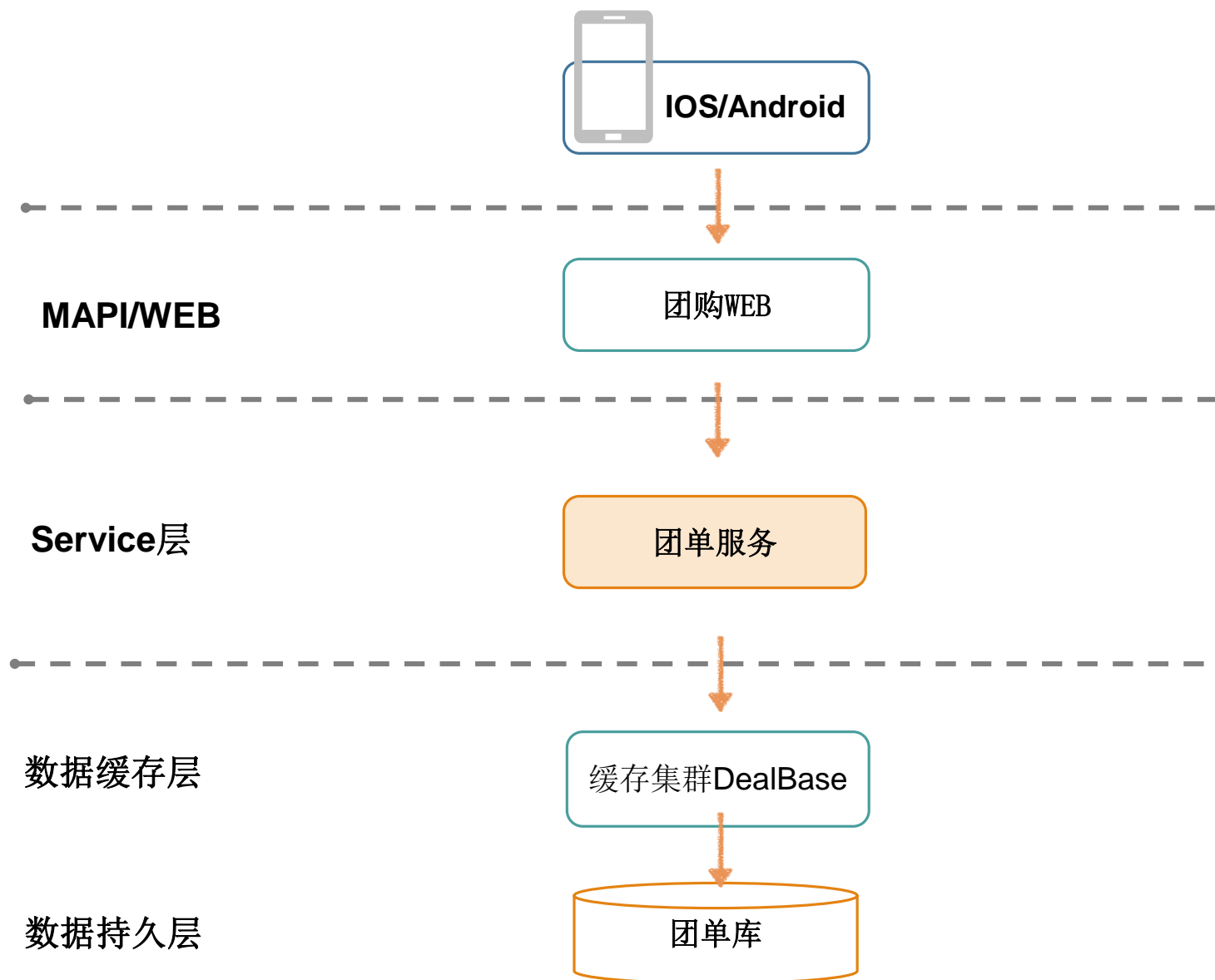
扩容是提升系统容量的最简单直接的办法，在系统没有瓶颈，可以水平扩展的前提下，扩容机器的数量为：

$$\text{扩容机器数} = \frac{\text{预估活动流量} \times \text{余量系数}}{\text{单机容量}} - \text{集群现有机器数}$$

团购系统架构演进

早期的团购系统

- 不管在线上还是在线上，做一次有效的压力测试比较困难。
- 根据木桶短板理论，不需要压力测试也可以知道当时的团单服务性能不会很好。



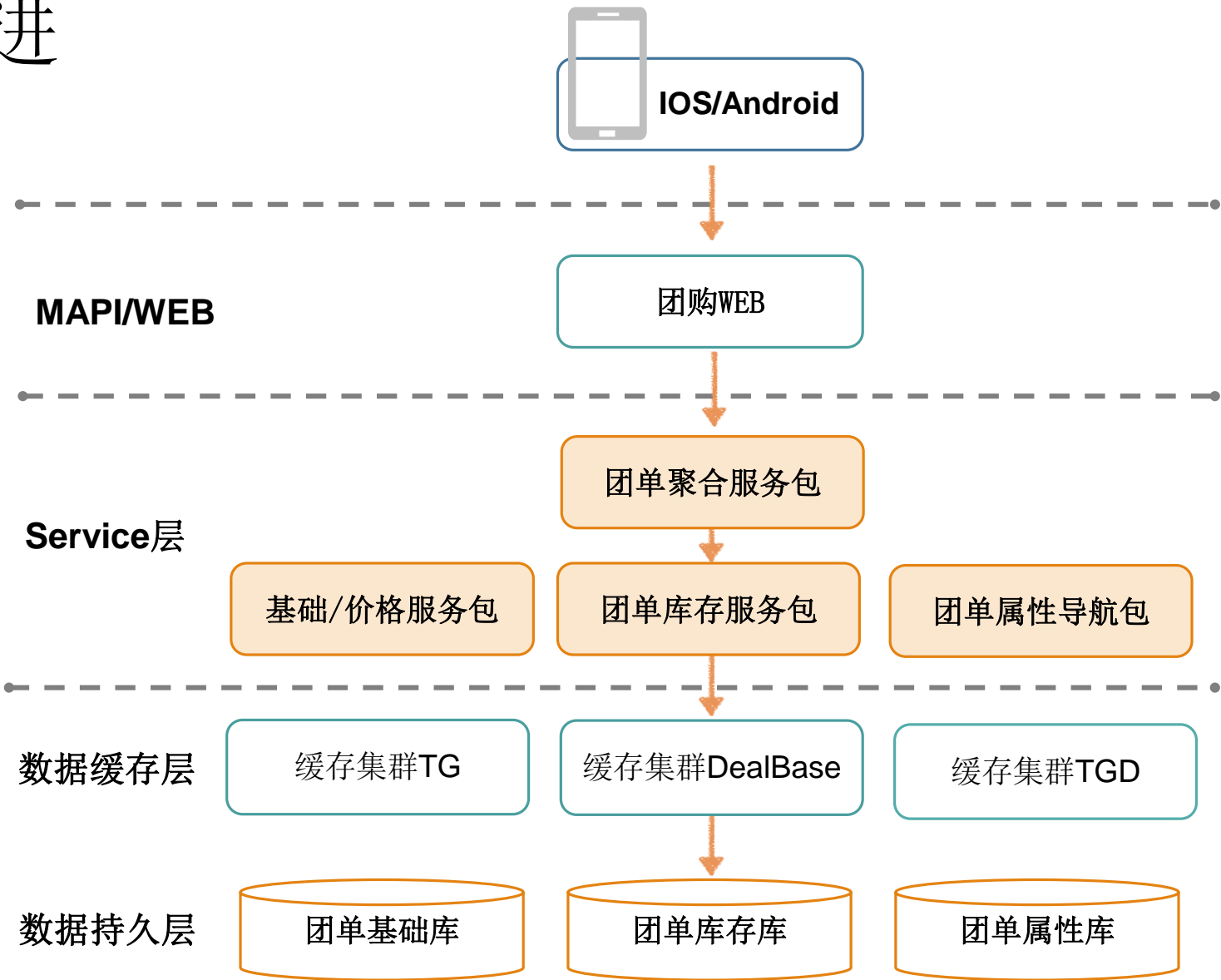
团购系统架构的演进

微服务化:

- 服务层拆包、消除出轨应用
- 逻辑耦合在一起的情况按服务化拆分出来，每个服务独立专注的做一件事情。

数据库拆分

- 拆分独立的团单属性库、库存库

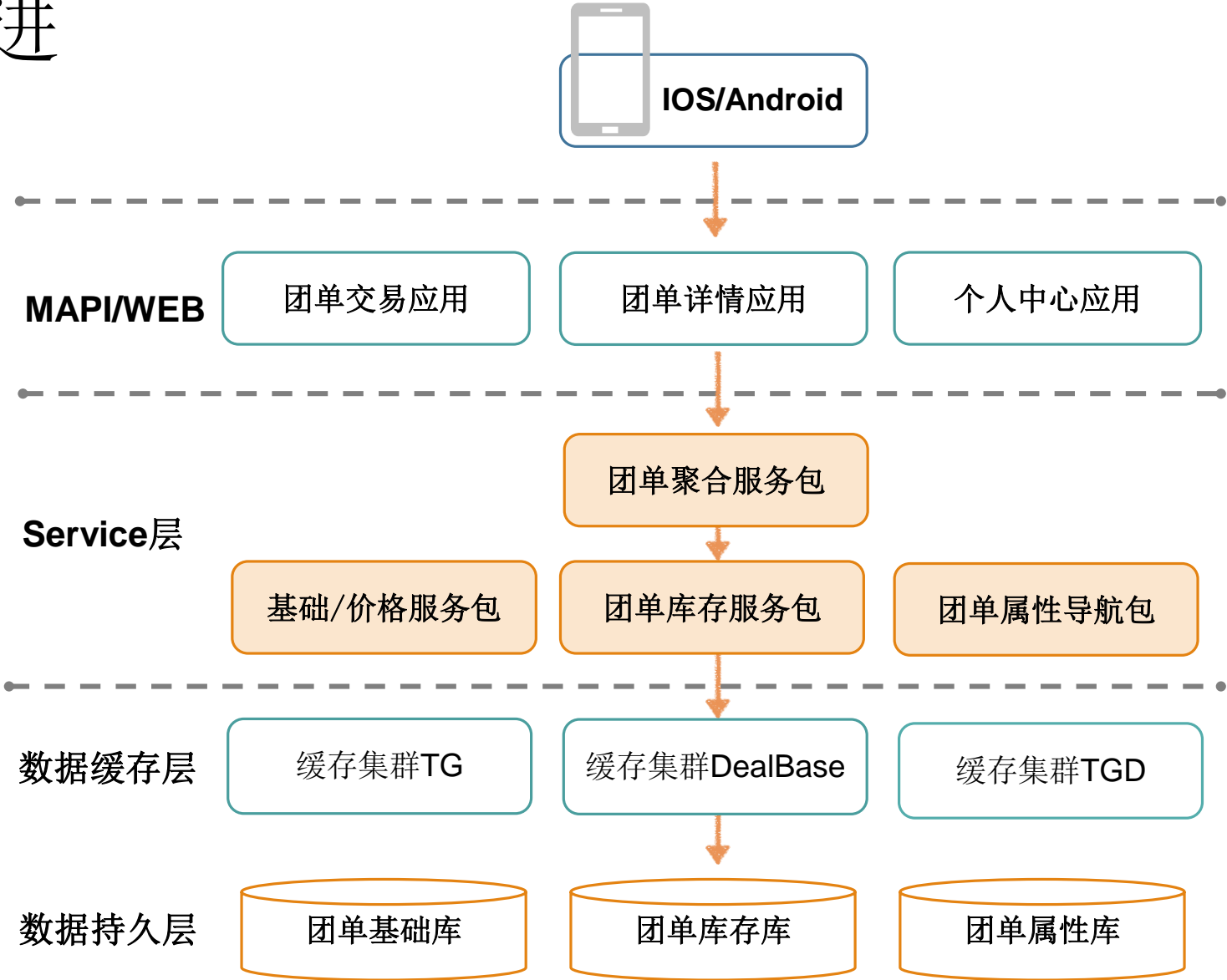


团购系统架构的演进

现在的团购系统

MAPI应用拆分

- MAPI系统大而全
- 影响系统容量、稳定性、研发效率
- 按照业务领域拆分成三个应用





复杂的系统架构如何扩容？

目录

大促活动前团购系统
流量预算和容量评估实践

01

背景介绍

02

活动流量预算

03

系统容量评估——压力测试策略

04

系统容量评估——压力测试过程

05

总结展望

大促活动流量模型

- 流量模型分析是容量评估的关键，只有清楚了系统的流量模型，才有可能进行准确的评估。
 - 平时的流量模型
 - 大促活动时的流量模型
 - 大促时核心路径的流量模型
 - 之后所讲到的流量分析和容量评估都是围绕这条核心路径展开

大促活动的核心路径

- 大促活动下，用户的行为是从活动落地页直接跳转到团购详情页，然后到支付流程并完成支付
- 在大促抢购活动一开始，商品系统的流量主要来自于团购详情页

1、活动页



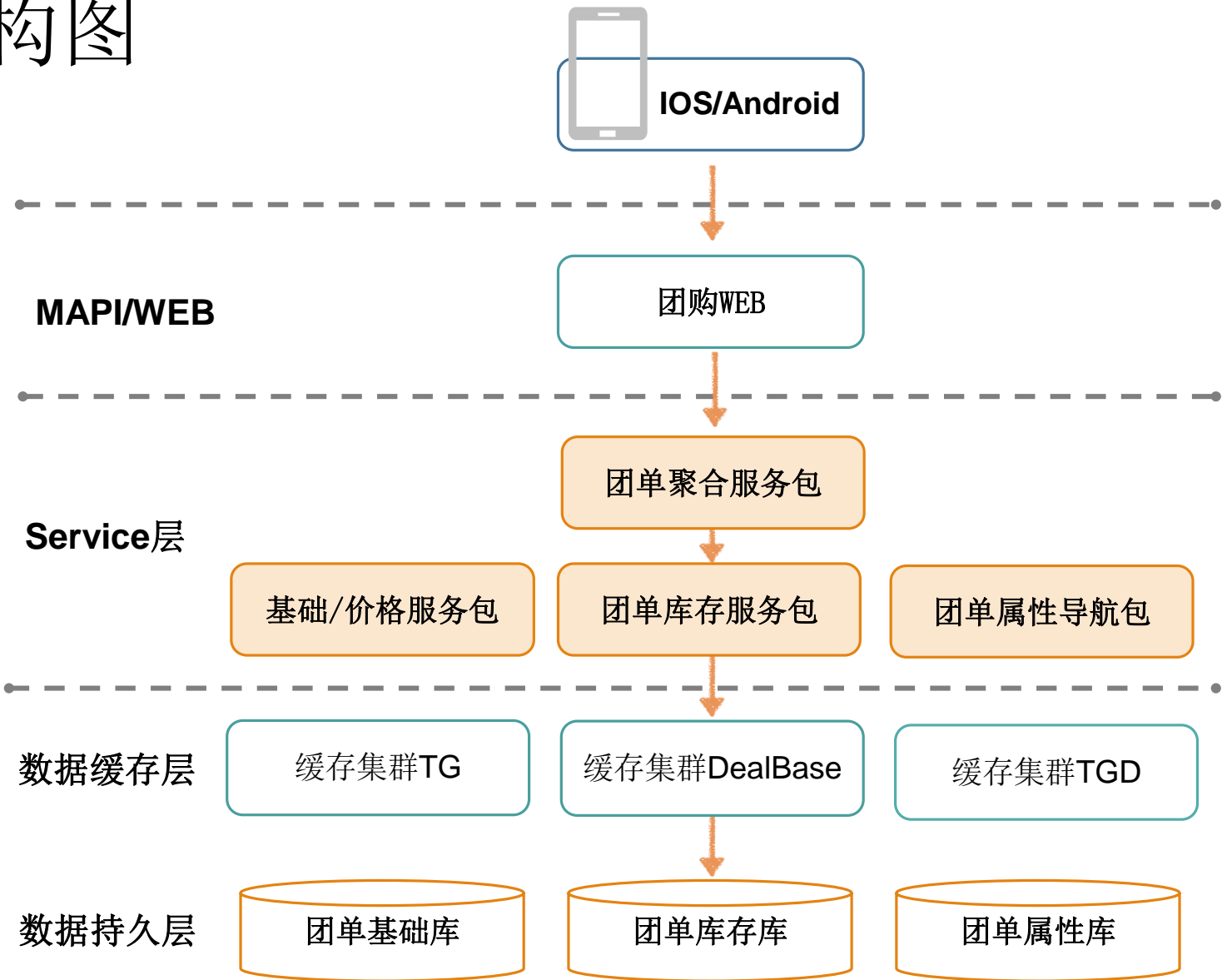
2、团购详情页



3、提交订单页



大促时核心路径架构图



活动流量预算

第一步：根据活动页每个按钮的位置 → 梳理APP Native页面每个接口的调用



团购详情应用有6个基础模块，占应用访问量的97%

核心路径

- 团购基本信息接口
- 团购详情须知接口
- 团购适用商户接口

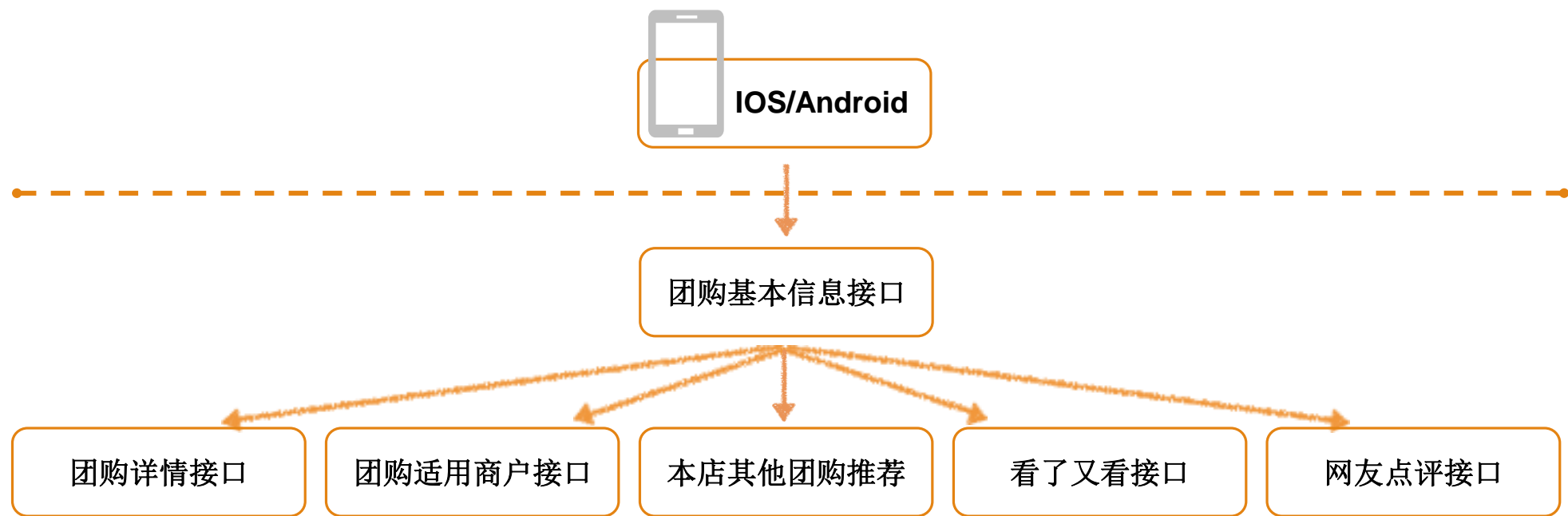
非核心路径

- 本店其他团购推荐接口
- 看了又看接口
- 网友点评接口



活动流量预算

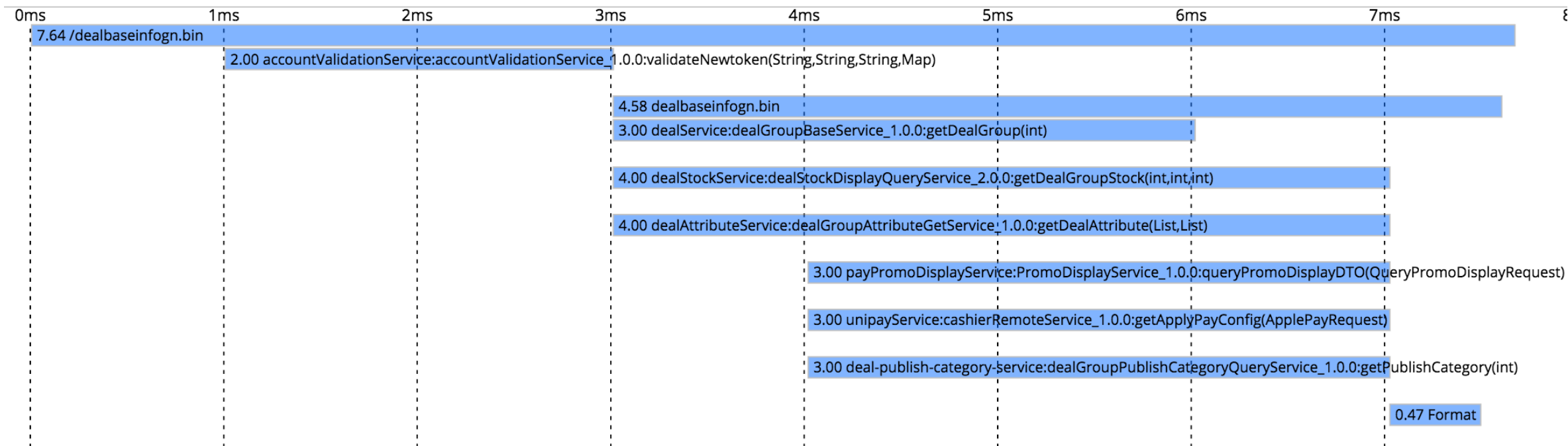
第二步：梳理接口之间的调用顺序以及接口内部的调用链



团购详情页6个基础模块的接口之间调用顺序

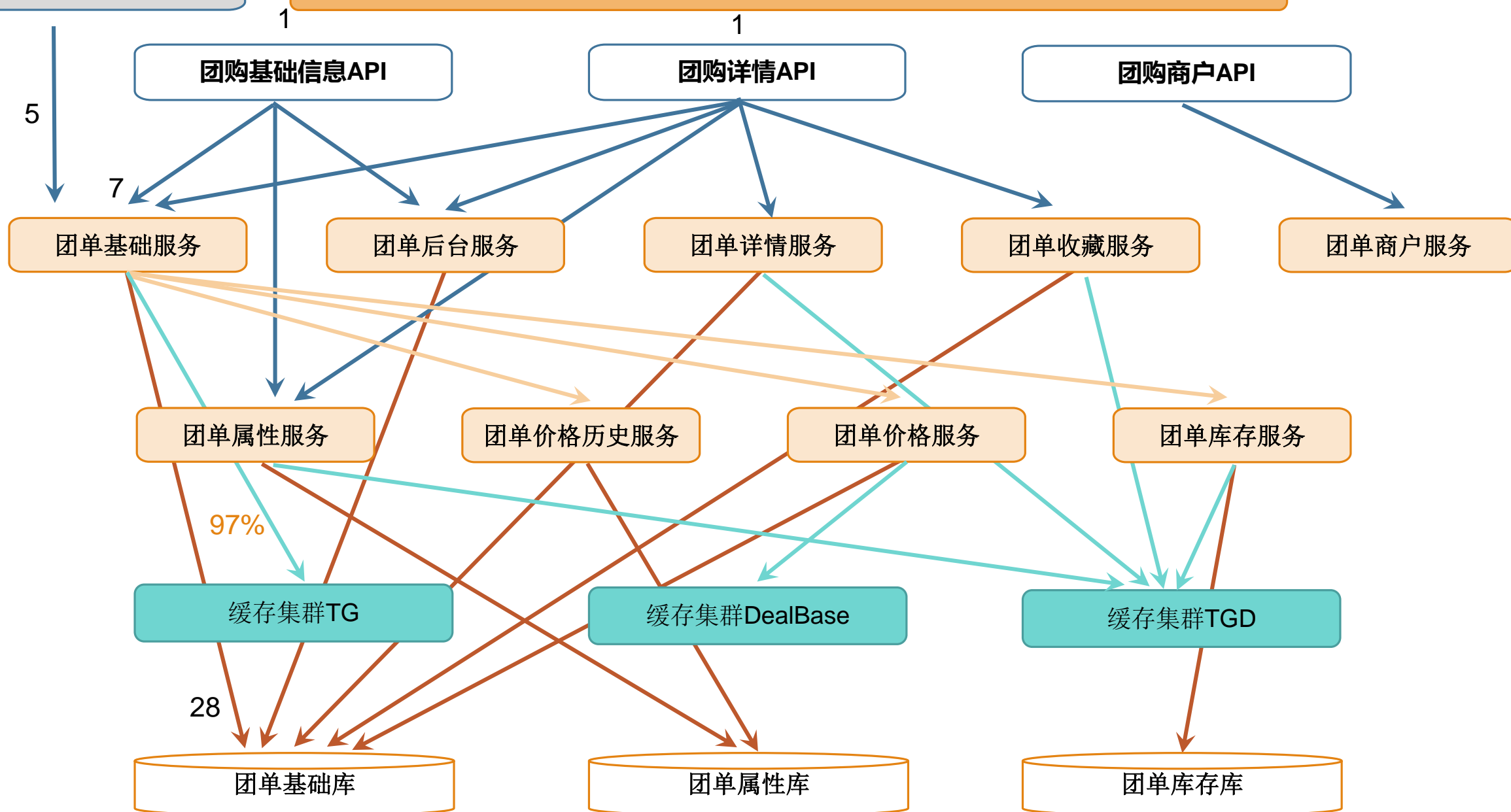
活动流量预算

第三步：通过代码分析和CAT堆栈，梳理接口依赖的服务及服务调用次数



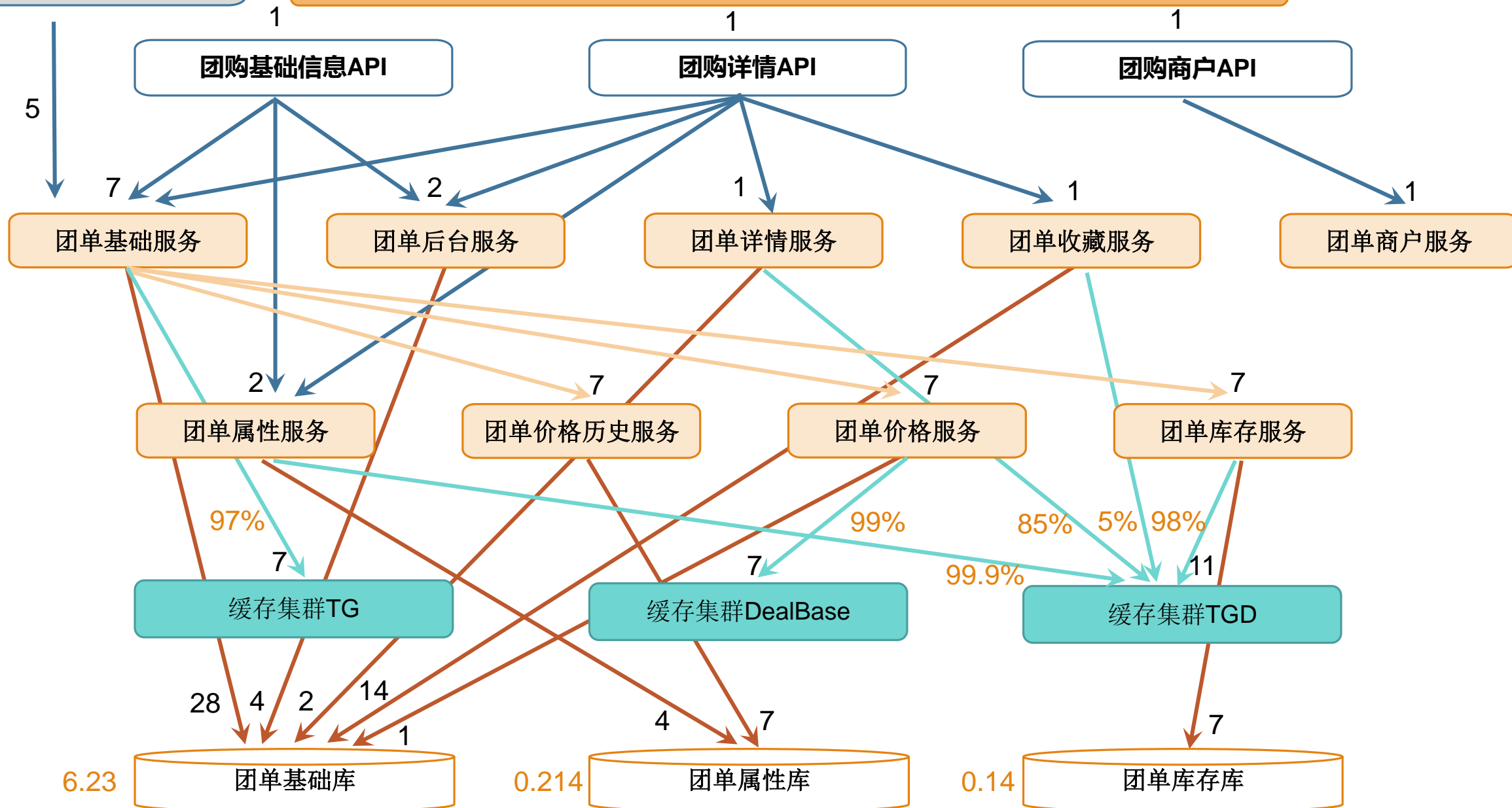
创建订单、支付等场景

第三步：梳理接口依赖的服务及服务调用次数



创建订单、支付等场景

第三步：梳理接口依赖的服务及服务调用次数



活动流量预算

在大促活动流量模型下，设详情页的流量为单位1，此时流量构成如下：

- 访问服务次数：

| 服务 | 访问次数 |
|---------|------|
| 团购基础服务 | 7 |
| 团购详情服务 | 1 |
| 团购价格服务 | 7 |
| 团购属性读服务 | 2 |
| 团购库存读服务 | 7 |
| 团购商户服务 | 1 |
| 团购后台服务 | 2 |
| 团购历史服务 | 7 |
| 团购收藏服务 | 1 |

活动流量预算

在大促活动流量模型下，设详情页的流量为单位1，此时流量构成如下：

- 访问cache次数：

| 缓存系统 | 访问次数 |
|-------------|------|
| memcached-A | 7 |
| memcached-B | 11 |
| memcached-C | 7 |

- 访问DB次数(通过分析cache命中率及各服务的读接口调用次数得出)：

| 数据库 | 访问次数(不考虑缓存) | 访问次数(考虑缓存) |
|-------|-------------|------------|
| 团购基础库 | 49 | 6.23 |
| 团购属性库 | 11 | 0.214 |
| 团购库存库 | 7 | 0.14 |

活动流量预算——扩容计划表

| 系统 | 应用 | 单机QPS | 活动流量预算 | 预计提供集群容量 | 当前机器数 | 需扩容机器数 |
|------|-------------|-------|--------|----------|-------|--------|
| MAPI | 团购基础应用 | ? | xxx | xxx | xxx | |
| | 团购详情应用 | ? | xxx | xxx | xxx | |
| | 团购交易应用 | ? | xxx | xxx | xxx | |
| | 个人中心应用 | ? | xxx | xxx | xxx | |
| | | | | | | |
| 商品系统 | 团购搜索服务 | ? | xxx | xxx | xxx | |
| | 团购基础服务 | ? | xxx | xxx | xxx | |
| | 团购详情服务 | ? | xxx | xxx | xxx | |
| | 。 。 。 。 。 。 | | | | | |

大促活动前的准备

$$\text{扩容机器数} = \frac{\text{预估活动流量} \times \text{余量系数}}{\text{单机容量}} - \text{集群现有机器数}$$

目录

大促活动前团购系统

流量预算和容量评估实践

01

背景介绍

02

活动流量预算

03

系统容量评估——压力测试策略

04

系统容量评估——压力测试过程

05

总结展望

系统容量评估——压力测试

理想

- 线上压测
- 系统全链路压测
- 一次压测解决问题

现实

- 线上压测有风险，成本高
- 时间紧（活动测试、版本发布和压力测试并行）
- 全链路压测依赖服务多，参与/周知人员多
- 全链路可以压出系统瓶颈，但是不能提供单机容量

保证压测数据有效性的基础上，做性价比最高的压力测试

系统容量评估——压力测试

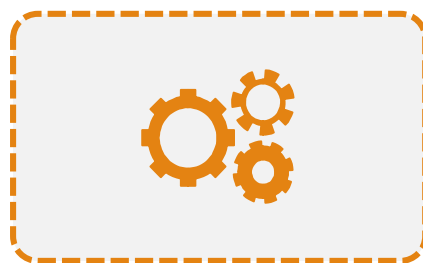
- 应用级压测方案，根据应用的类型，分层压测
 - MAPI-WEB
 - Pigeon Service聚合服务
 - Pigeon Service基础服务
 - 数据库
- 根据实际场景特点，选择压测环境
 - 区分读场景、写场景

系统容量评估——压力测试



压测策略

- 压测目的
- 压测环境
- 风险成本



压测过程

- 压测方法
- 压测场景
- 压测数据



压测结论

- 数据采集
- 数据分析
- 监控告警

系统容量评估——压测目的



单机容量

应用的单机容量是多少？

集群容量

是否可以通过水平扩容来提升集群容量，系统瓶颈在哪？



DB容量

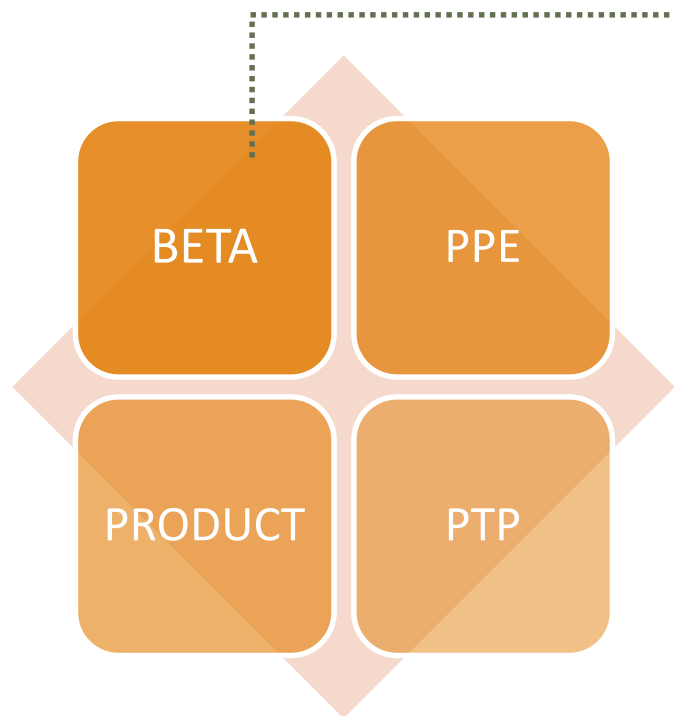
数据库能承担的压力是多少？

监控告警

监控指标应该如何设置，若超过阈值告警机制是否生效？



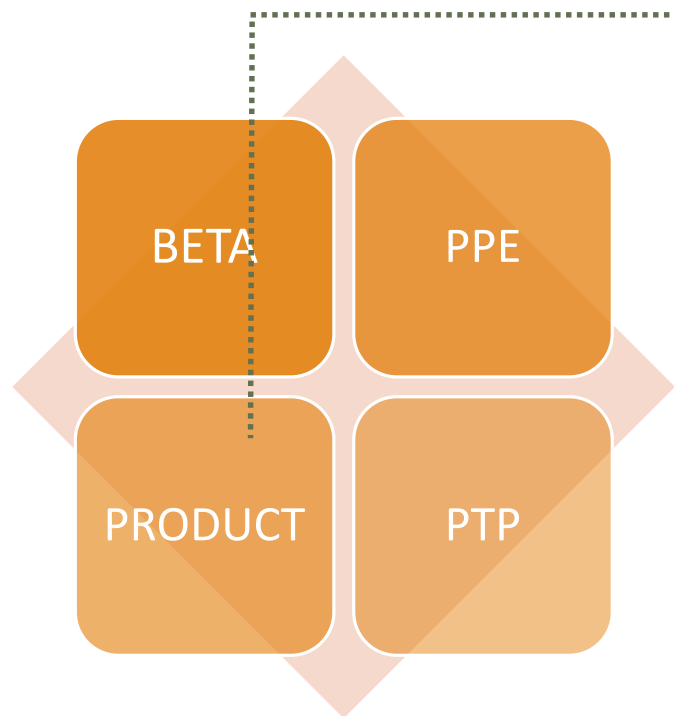
系统容量评估——压测环境



□ BETA / PPE环境

- 数据可靠性：硬件环境、数据等与线上差异较大；
- 稳定性：被测服务和依赖服务容易被压挂，影响功能测试；
- 易用性：支持常用压测工具和方法，压测脚本在本地执行；
- 局限性：只能评估单机容量，无法对集群进行压测；

系统容量评估——压测环境



□ 线上环境

- **数据可靠性**：数据可靠，无环境差异，可以压测集群；
- **稳定性**：线上被测服务或依赖服务压挂风险很高；
- **易用性**：需运维辅助操作，在线上摘机器，或者拷贝流量；
- **局限**：有一定的风险，不是所有类型的应用都适合；

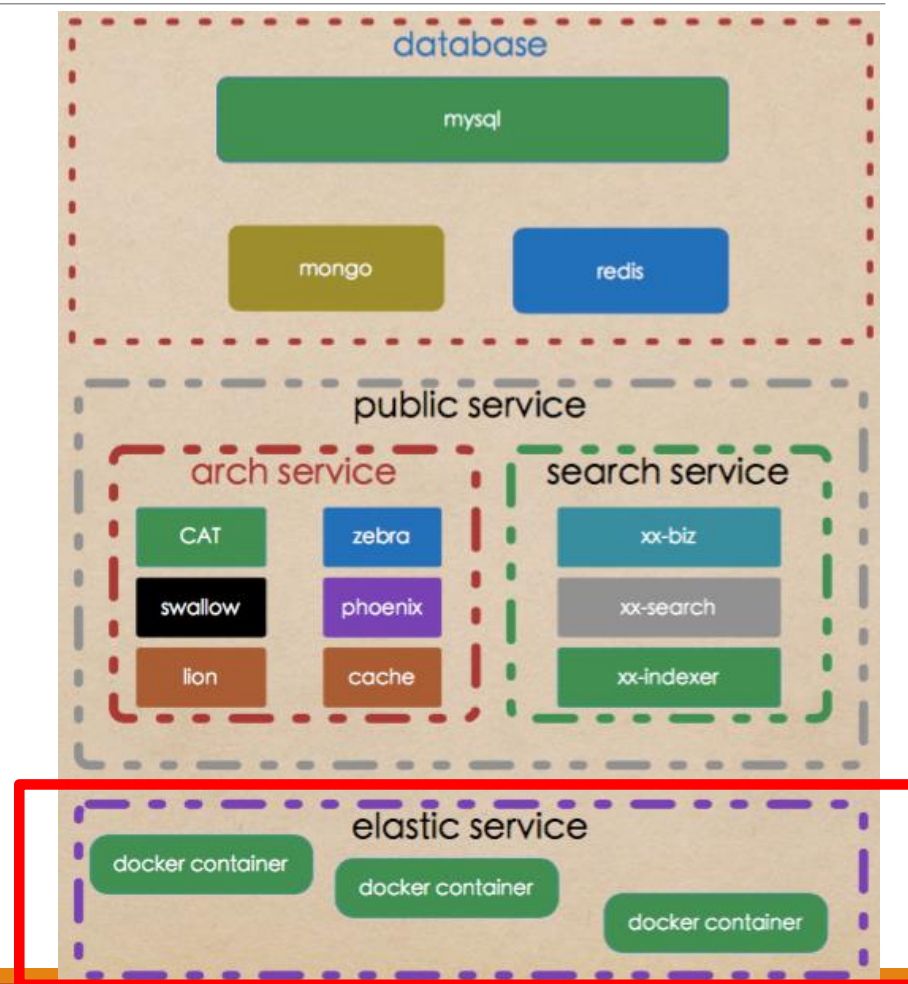
容量评估方法——压测环境

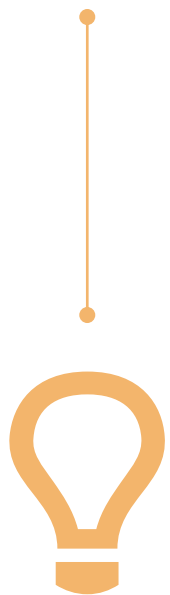
复制一套线上环境和数据，搭建性能测试环境？

- 机器数量有限
- 维护成本很高
- 资源利用率低

容量评估方法——PTP性能测试环境

- 机器数量有限 → 基于Docker的虚拟机器池
- 维护成本很高 → 一键部署被测应用
- 资源利用率低 → 定期环境清理





如何部署依赖的服务？

容量评估方法——PTP性能测试环境

- 智能依赖模块
 - ❑ 直接依赖服务
 - ❑ 间接依赖服务
 - ❑ 数据库依赖
 - ❑ 公共服务
 - ❑ 手动添加



容量评估方法——PTP性能测试环境

• 智能依赖模块





直接获取的依赖服务不全怎么办？

容量评估方法——PTP性能测试环境

回流依赖模块



- ❑ 直接执行测试脚本
- ❑ 获取缺失的依赖
- ❑ 正式执行测试脚本
- ❑ Jmeter聚合报告
- ❑ 循环调用部署模块
- ❑ 展示实时性能数据
- ❑ CAT Heartbeat实时报表
- ❑ 继续回流分析
- ❑ 性能监控工具VM等
- ❑ 直至无依赖缺失

监控系统CAT: Problem

- **数据可靠性：**

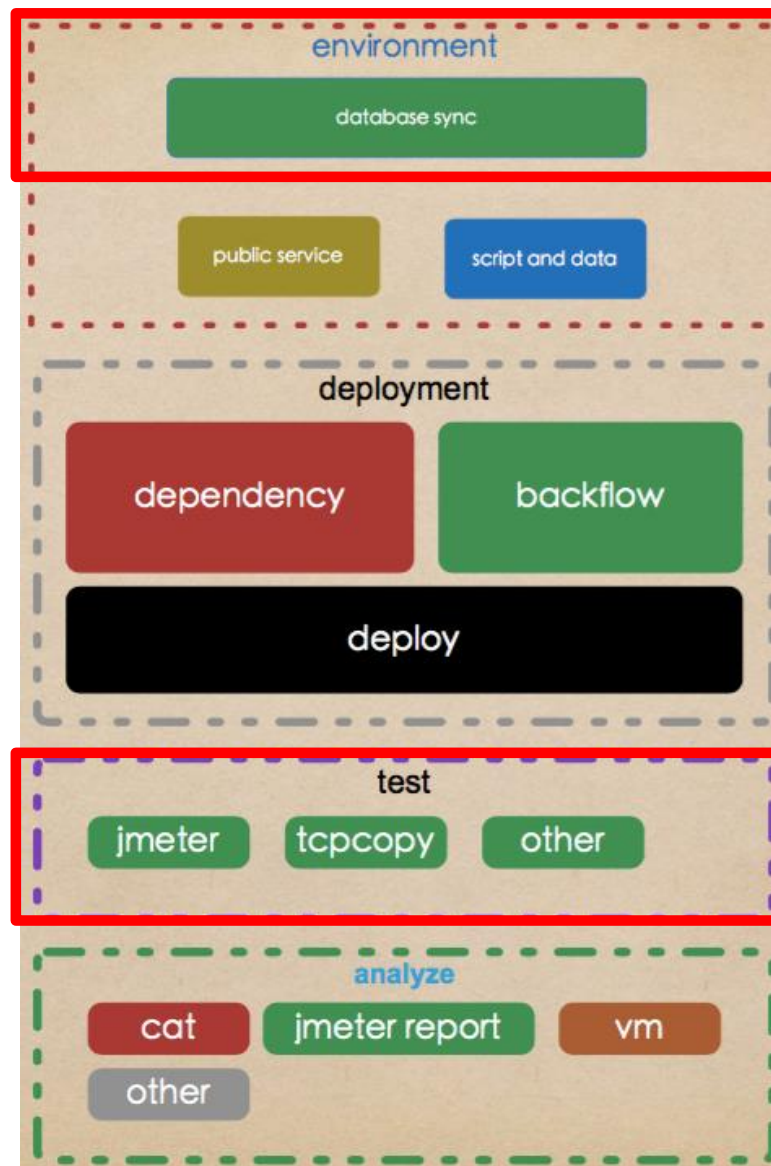
- 硬件环境灵活配置 ✓
- 数据库可以随时同步线上数据 ✓
- 数据较可靠 ✓
- 施压机和被测应用都可以是集群 ✓

- **稳定性：**

- 专职专用，不会影响正常的功能测试 ✓
- 自动部署依赖应用的稳定版本 ✓
- 支持泳道配置，可以同时压测多个服务 ✓

- **易用性：**

- 支持jmeter、tcpcopy等压测方法 ✓
- 施压机和被测应用都可以是集群 ✓



目录

大促活动前团购系统

流量预算和容量评估实践

01

背景介绍

02

活动流量预算

03

系统容量评估——压力测试策略

04

系统容量评估——压力测试过程

05

总结展望

系统容量评估——压力测试



压测策略

- 压测目的
- 压测环境
- 风险成本



压测过程

- 压测方法
- 压测场景
- 压测数据



压测结论

- 数据采集
- 数据分析
- 监控告警

系统容量评估——压测方法

□ 线上压测方法——逐步减小集群规模

- 通过逐台摘机器的方法，使得单台机器的访问量不断提升，来达到压力测试的目的。需要运维人员的密切配合。
- 优点：应用无需是读写接口分离的
- 缺点：1. 逐个摘机器有风险 2. 对流量规模有要求，不能太小，否则压不到瓶颈

系统容量评估——压测方法

□ 线上压测方法——线上Tcpcopy

- 在集群中挑选A、B两台机器，A为被压机，B为施压机。
- 将A配置一个泳道（机器不对外提供服务），通过Tcpcopy将B的流量逐渐放大至A。
- 优点：

流量比例、风险小，可以随时修改放大倍数

- 缺点：
 - 1、要求被压的服务只能是纯读服务，不能有写接口，否则会带来脏数据
 - 2、Tcpcopy工具限制，放大40倍会挂掉，若线上流量太小。则需要首先摘掉部分机器以提高单机的QPS

系统容量评估——压测方法

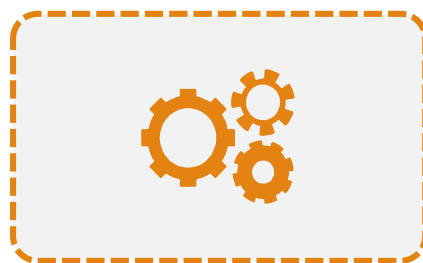
- 线下压测方法——性能测试平台
 - 模拟线上机器配置、JVM配置
 - 模拟线上接口请求比例
 - 模拟线上接口请求参数分布
 - 模拟线上缓存命中率

系统容量评估——压力测试



压测策略

- 压测目的
- 压测环境
- 风险成本



压测过程

- 压测方法
- 压测场景
- 压测数据



压测结论

- 数据采集
- 数据分析
- 监控告警

系统容量评估——压测场景



实例一

- 团购基础查询服务提供了两种查询方式：单个查询和批量查询
- 设计测试场景时需要考虑两种接口的比例分布
- 使用jmeter比例控制器做控制

| Name | Total | Failure | Failure% | Sample Link | Min(ms) | Max(ms) | Avg(ms) | 95Line(ms) | 99.9Line(ms) | Std(ms) | QPS | Percent% |
|--|-----------|---------|----------|--------------------------|---------|---------|---------|------------|--------------|---------|-------|----------|
| TOTAL | 3,837,734 | 0 | 0.0000% | Log View | 0 | 970 | 5.3 | - | - | 9.6 | 0.0 | 100.00% |
| dealService:dealGroupBaseService_1.0.0:getDealGroup(int) | 1,724,345 | 0 | 0.0000% | Log View | 0 | 970 | 4.2 | 11.3 | 25.1 | 5.8 | 479.0 | 44.93% |
| dealService:dealGroupBaseService_1.0.0:multiGetDealGroup(List) | 1,364,444 | 0 | 0.0000% | Log View | 0 | 929 | 7.4 | 19.6 | 200.0 | 14.0 | 379.0 | 35.55% |
| dealService:dealBaseService_1.0.0:getDeal(int) | 424,333 | 0 | 0.0000% | Log View | 0 | 829 | 3.8 | 8.1 | 19.0 | 4.8 | 117.9 | 11.06% |
| dealService:dealBaseService_1.0.0:multiGetDeal(List) | 324,612 | 0 | 0.0000% | Log View | 0 | 720 | 4.3 | 10.5 | 35.2 | 5.8 | 90.2 | 8.46% |

系统容量评估——压测场景



实例二

- 团购基础查询服务的批量接口参数个数
 - 需要设计调用批量查询接口时传入的团单id个数
-

系统容量评估——压测场景



实例三

- 针对对应用性能有影响的参数类型、逻辑分支设计场景
 - 例如团购销量查询服务区分品类
 - 结婚团单需要聚合销量、丽人团单需要分城市销量
 - 不同的逻辑分支，查询不同的缓存和数据库
-

系统容量评估——压测场景



实例四

- 针对开关和降级场景设计场景
 - 例如团购价格查询服务有一个“诚信旁路”开关
 - 对“诚信旁路”开关打开或关闭的情况分别进行压力测试
 - 大促期间如果必要可以把“诚信旁路”开关关闭
-

系统容量评估——压测数据

□ 压力测试数据的构造

- 线上流量tcpdump，到性能测试环境做流量翻倍回放
- 采取MAPI Nginx access log回放机制，对用户真实访问行为进行回放
- 人工构造数据

系统容量评估——压测数据

实例一

- **被测应用：**团购详情WEB应用（纯读服务）
- **压测环境：**性能测试环境
- **压测数据：**tcpdump复制线上流量在性能环境做回放
- **压测方法：**
 - 选择合适的时间段（流量不要太低）
 - dump流量时间足够长，流量翻倍时数据离散度需满足要求



系统容量评估——压测数据

实例二

- **被测应用：**个人中心WEB应用（读+写）
- **压测环境：**性能测试环境
- **压测数据：**nginx access log
- **压测方法：**
 - 将最近通过app的访问nginx access log进行分类（读、写）
 - 线上到线下token的解析问题
 - 重复写请求报错的问题



系统容量评估——压测数据

```
192.168.216.185 10.66.40.77 [06/Sep/2016:13:10:29 +0800] 1473138629.032+1799879 "GET /dealdetailinfogn.bin?id=200030115&shopid=3169789&token=7019c1aee56420793d570659246778e1a767b0a0dff6bb8a0bde44305cb05c50 HTTP/1.1" 200 1799879 0.066 1056 "-" "MApi 1.1 (com.dianping.v1 9.0.0 null HUAWEI_P7-L07; Android 4.4.2)" "180.166.152.90, 192.168.5.8, 10.66.40.77" - -
192.168.216.185 10.66.40.185 [06/Sep/2016:13:10:29 +0800] 1473138629.070+1799871 "GET /moredealslistgn.bin?categorykeys=70%2C7001&lng=121.415846&shopid=3169789&dealgroupid=200030115&token=7019c1aee56420793d570659246778e1a767b0a0dff6bb8a0bde44305cb05c50&lat=31.21796&cityid=1 HTTP/1.1" 200 1799871 0.110 880 "-" "MApi 1.1 (com.dianping.v1 9.0.0 null HUAWEI_P7-L07; Android 4.4.2)" "180.166.152.90, 192.168.5.8, 10.66.40.185" - -
192.168.216.185 10.66.40.77 [06/Sep/2016:13:10:29 +0800] 1473138629.445+1799881 "GET /dealdetailinfogn.bin?id=200030115&shopid=3169789&token=7019c1aee56420793d570659246778e1a767b0a0dff6bb8a0bde44305cb05c50 HTTP/1.1" 200 1799881 0.022 1056 "-" "MApi 1.1 (com.dianping.v1 9.0.0 null HUAWEI_P7-L07; Android 4.4.2)" "180.166.152.90, 192.168.5.8, 10.66.40.77" - -
192.168.216.185 10.66.40.77 [06/Sep/2016:13:10:56 +0800] 1473138656.006+1799887 "GET /dealbaseinfogn.bin?id=200030116&cityid=1&shopid=0&isgoodshop=0&token=7019c1aee56420793d570659246778e1a767b0a0dff6bb8a0bde44305cb05c50&lat=31.21796&lng=121.415846 HTTP/1.1" 200 1799887 0.109 832 "-" "MApi 1.1 (com.dianping.v1 9.0.0 null HUAWEI_P7-L07; Android 4.4.2)" "180.166.152.90, 192.168.5.8, 10.66.40.77" - -
192.168.216.185 10.66.40.77 [06/Sep/2016:13:10:56 +0800] 1473138656.216+1799889 "GET /bestshopgn.bin?lng=121.415846&shopid=0&dealgroupid=200030116&token=7019c1aee56420793d570659246778e1a767b0a0dff6bb8a0bde44305cb05c50&lat=31.21796&cityid=1 HTTP/1.1" 200 1799889 0.022 304 "-" "MApi 1.1 (com.dianping.v1 9.0.0 null HUAWEI_P7-L07; Android 4.4.2)" "180.166.152.90, 192.168.5.8, 10.66.40.77" - -
192.168.216.185 10.66.40.185 [06/Sep/2016:13:10:56 +0800] 1473138656.371+1799891 "GET /dealdetailinfogn.bin?id=200030116&shopid=3169789&token=7019c1aee56420793d570659246778e1a767b0a0dff6bb8a0bde44305cb05c50 HTTP/1.1" 200 1799891 0.039 1776 "-" "MApi 1.1 (com.dianping.v1 9.0.0 null HUAWEI_P7-L07; Android 4.4.2)" "180.166.152.90, 192.168.5.8, 10.66.40.185" - -
192.168.216.185 10.66.40.77 [06/Sep/2016:13:10:56 +0800] 1473138656.372+1799894 "GET /bestshoppreviewgn.bin?lng=121.415846&shopid=3169789&lat=31.21796&dealid=200030116&cityid=1 HTTP/1.1" 200 1799894 0.031 624 "-" "MApi 1.1 (com.dianping.v1 9.0.0 null HUAWEI_P7-L07; Android 4.4.2)" "180.166.152.90, 192.168.5.8, 10.66.40.77" - -
```

Nginx access日志

系统容量评估——压测数据

实例三

- **被测应用：**团购基础信息服务
- **压测环境：**性能测试环境
- **压测数据：**人工构造数据（csv文件）
- **压测方法：**
 - 人工构造的数据应模拟线上的缓存命中率
 - 缓存命中率比线上环境偏低，需要进行缓存预热
 - 缓存命中率比线上环境偏高，需要尽量模拟线上团单的离散程度



系统容量评估——压测工具

□ 压力测试工具的选择

- Jmeter + Http
- Jmeter + Pigeon Client
- arch/test benchmark
- PTP性能测试平台 (Jmeter/Tcpcopy)
- PTEST(支持高并发、万级QPS)

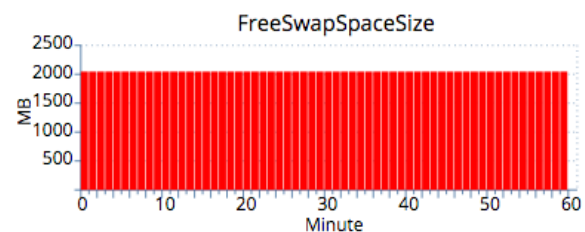
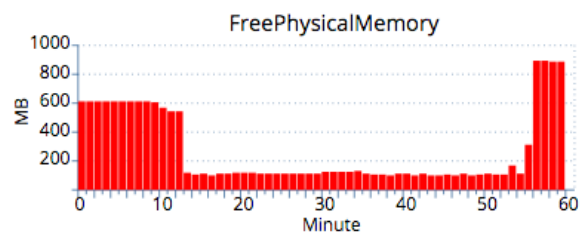
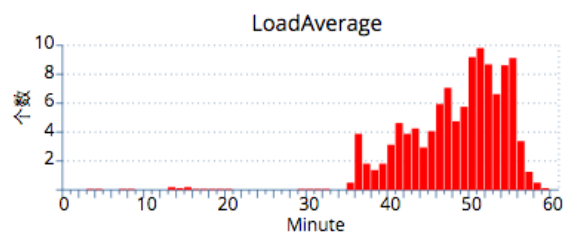
系统容量评估——压测结果采集

□ 压力结果数据的采集

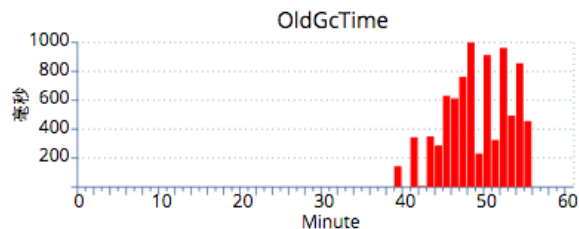
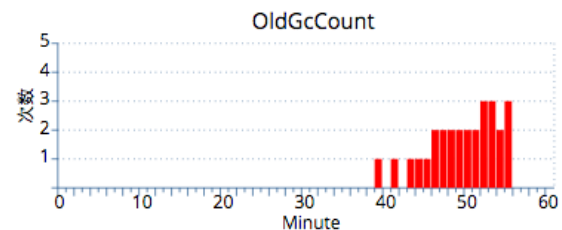
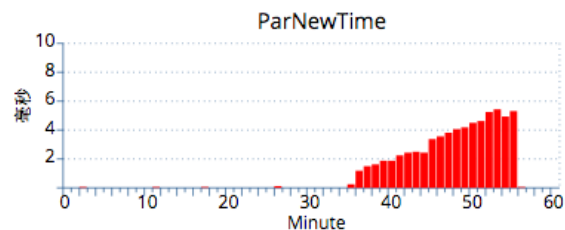
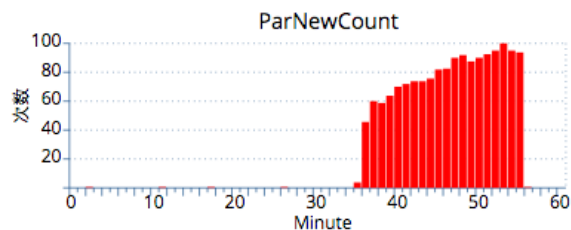
- CAT Heartbeat / Transaction
- PTP性能测试平台监控报告
- PTEST平台监控报告
- Jconsole/ VisualVM （推荐，集成多个JVM命令的可视化工具）
- JVM命令：jstack/jstat/jmap

系统容量评估——压测结果采集

System Info



GC Info



扩容计划表

| 系统 | 应用 | 单机QPS | 活动流量预算 | 预计提供集群容量 | 当前机器数 | 需扩容机器数 |
|------|-------------|-------|--------|----------|-------|--------|
| MAPI | 团购基础应用 | xxx | xxx | xxx | xxx | |
| | 团购详情应用 | xxx | xxx | xxx | xxx | |
| | 团购交易应用 | xxx | xxx | xxx | xxx | |
| | 个人中心应用 | xxx | xxx | xxx | xxx | |
| | | | | | | |
| 商品系统 | 团购搜索服务 | xxx | xxx | xxx | xxx | |
| | 团购基础服务 | xxx | xxx | xxx | xxx | |
| | 团购详情服务 | xxx | xxx | xxx | xxx | |
| | 。 。 。 。 。 。 | | | | | |

系统容量评估——DB容量

- 压测目的：获取数据库最高可以承受的QPS
- 压测环境：性能测试环境（全量同步线上数据库）
- 压测数据：平日10点高峰流量，通过分析sql访问日志，从中获取各类sql的比例与in参数的分布
- 压测方法：通过Jmeter + JDBC Request执行压测

系统容量评估——DB容量

主库压测数据:

| 数据库 | 流量类型 | 压测类型 | 峰值 | 读写比 |
|--------------|-----------|------|-------|-------|
| 团购属性库 Master | 平日10点高峰流量 | 读+写 | 12w | 10: 3 |
| 团购基础库 Master | 平日10点高峰流量 | 读+写 | 7w | 60: 1 |
| 团购库存库 Master | 平日10点高峰流量 | 读+写 | 15.8w | 11: 1 |

从库压测数据:

| 数据库 | 流量类型 | 压测类型 | 峰值 | 主从读流量比 |
|-------------|-----------|------|-------|--------|
| 团购属性库 Slave | 平日10点高峰流量 | 读 | 12.3w | 13: 10 |
| 团购基础库 Slave | 平日10点高峰流量 | 读 | 12.3w | 17: 10 |

容量评估方法——缓存容量

| 集群名 | 节点数量 | 容量 | 应用 | 流量占比 | 实际容量 |
|--------------------|------|------|--|------|------|
| memcached-tgdeal | 3 | 24w | deal-price-query-service | 100% | 24w |
| memcached-dealbase | 3 | 24w | deal-base-service | 99% | 24w |
| memcached-tuangou | 13 | 104w | deal-detail-service deal-attribute-get-service deal-stock-query-service deal-favorite-service | 72% | 75w |

性能优化

- 性能优化没有银弹
- 架构的设计（微服务拆分、异步化改造）
- 业务的裁减（不必要的查询、服务接口批量调用）
- 活动前期避免性能优化引入的新的性能不确定性

降级和流控方案

- Lion开关集合
- 关闭“诚信旁路”分支调用
- 关闭非核心路径接口功能
- 流量超过阈值随机失败
- 静态化
- 降级预案有多种，大部分都用不到，但是要定期验证降级预案的有效性

目录

大促活动前团购系统

流量预算和容量评估实践

01

背景介绍

02

活动流量预算

03

系统容量评估——压力测试策略

04

系统容量评估——压力测试过程

05

总结展望

总结

- 活动流量预算：从上到下评估流量
- 系统容量评估：从下到上提供能力
- 保证大促期间核心路径用户体验

展望

- 简单化
- 标准化
- 流程化
- 自动化

Thank You
